

**Amendments to the Claims:**

This listing of claims will replace all prior versions and listings of claims in the application:

**Listing of Claims:**

Claim 1 (currently amended): A computer system comprising a memory portion containing an encrypted data file and an operating system comprising a kernel to use ~~system-unique data~~ a unique system-identifier to verify a user to control access to the encrypted data file, wherein the kernel comprises a virtual node (a) to decrypt an encrypted directory entry to determine a location of the encrypted data file and (b) to decrypt the encrypted data file to access data file contents contained therein.

Claim 2 (previously presented): The computer system of claim 1, wherein the kernel comprises an encryption engine to encrypt clear data files to generate cipher data files, the encryption engine also to decrypt the cipher data files to generate the clear data files.

Claim 3 (previously presented): The computer system of claim 2, wherein the memory portion is coupled to the encryption engine to store the cipher data files.

Claim 4 (previously presented): The computer system of claim 2, wherein the encryption engine is to encrypt the clear data files and decrypt the cipher data files according to a symmetric key encryption algorithm.

Claim 5 (previously presented): The computer system of claim 4, wherein the symmetric key encryption algorithm is based on a block cipher.

Claim 6 (previously presented): The computer system of claim 5, wherein the symmetric key encryption algorithm comprises Rijndael algorithm.

Claim 7 (previously presented): The computer system of claim 6, wherein the symmetric key encryption algorithm uses a block size of 128 bits, 192 bits, 256 bits, 512 bits, 1024 bits, or 2048 bits.

Claim 8 (previously presented): The computer system of claim 6, wherein the symmetric key encryption algorithm uses a key length of 128 bits, 192 bits, 256 bits, 512 bits, 1024 bits, or 2048 bits.

Claim 9 (previously presented): The computer system of claim 5, wherein the symmetric key encryption algorithm comprises a DES algorithm.

Claim 10 (previously presented): The computer system of claim 5, wherein the symmetric key encryption algorithm comprises a Triple-DES algorithm.

Claim 11 (previously presented): The computer system of claim 5, wherein the symmetric key encryption algorithm comprises an algorithm selected from the group consisting of IDEA, Blowfish, Twofish, and CAST-128.

Claim 12 (previously presented): The computer system of claim 1, wherein the kernel comprises a UNIX operating system.

Claim 13 (previously presented): The computer system of claim 12, wherein the UNIX operating system is a System V-Revision.

Claim 14 (previously presented): The computer system of claim 1, wherein the memory portion comprises a first logical protected memory to store encrypted data files and a second logical protected memory to store encrypted key data.

Claim 15 (previously presented): The computer system of claim 14, further comprising an encryption key management system to control access to the encrypted data files and the encrypted key data.

Claim 16 (previously presented): The computer system of claim 15, wherein the encryption key management system comprises a key engine, the key engine to receive a pass key and a data file name to generate an encrypted data file name key, the key engine also to use the encrypted data file name key and the data file contents to generate an encrypted data file contents key, the key engine also to encrypt the data file contents with the encrypted data file contents key to generate encrypted data file contents and to encrypt the data file name with the encrypted data file name key to generate an encrypted data file name.

Claim 17 (previously presented): The computer system of claim 16, wherein the encryption key management system is to store the encrypted data file name, wherein the data file name is associated with the encrypted data file contents.

Claim 18 (previously presented): The computer system of claim 17, wherein the encryption key management system is also to grant access to a data file if a corresponding access permission of the data file is a predetermined value.

Claim 19 (previously presented): The computer system of claim 1, further comprising a secondary device coupled to the memory, wherein the secondary device stores the encrypted data file and is accessed using a file abstraction.

Claim 20 (previously presented): The computer system of claim 19, wherein the secondary device is a backing store.

Claim 21 (previously presented): The computer system of claim 19, wherein the secondary device is a swap device.

Claim 22 (previously presented): The computer system of claim 19, wherein the secondary device comprises an interface port comprising a socket connection.

Claim 23 (previously presented): The computer system of claim 22, wherein the socket connection comprises a computer network.

Claim 24 (previously presented): The computer system of claim 23, wherein the computer network comprises the Internet.

Claim 25 (previously presented): The computer system of claim 17, wherein the encryption key management system is also to encrypt a pathname to the encrypted data file, and to decrypt the pathname to the encrypted data file when retrieving the encrypted data file contents.

Claim 26 (previously presented): A computer system comprising:

- a. a first device having an operating system kernel and a directory structure with directory information comprising encrypted data file names and corresponding encrypted data file locations for accessing encrypted data files within a file system, the operating system kernel to decrypt the encrypted data file names and encrypted data file locations using one or more encryption keys to recover clear data corresponding to the data file names, data file locations, and data files, the operating system kernel comprising a virtual node to encrypt the clear data using the one or more encryption keys to generate cipher data corresponding to the directory information and encrypted data files;
- b. a key generator to generate the one or more encryption keys from identifiers unique to the computer system and unique to encrypted data files on the computer system; and
- c. a second device coupled to the first device to exchange cipher data with the first device.

Claim 27 (previously presented): The computer system of claim 26, wherein the operating system kernel is to encrypt clear data and decrypt cipher data using a symmetric algorithm.

Claim 28 (original): The computer system of claim 27, wherein the symmetric algorithm comprises a block cipher.

Claim 29 (original): The computer system of claim 28, wherein the block cipher comprises a Rijndael algorithm.

Claim 30 (previously presented): The computer system of claim 29, wherein one of the one or more encryption keys comprises at least 1024 bits.

Claim 31 (original): The computer system of claim 26, wherein the second device comprises a backing store.

Claim 32 (original): The computer system of claim 26, wherein the second device comprises a swap device.

Claim 33 (previously presented): The computer system of claim 26, wherein the second device forms part of a communications channel.

Claim 34 (original): The computer system of claim 33, wherein the communications channel comprises a network.

Claim 35 (original): The computer system of claim 34, wherein the network comprises the Internet.

Claim 36 (previously presented): A method of storing an encrypted data file in a computer file system having a directory, the method comprising:

- a. receiving a clear data file having a name; and
- b. executing kernel code in an operating system, the kernel code comprising a virtual node comprising drivers to encrypt the clear data file to generate an encrypted data file using a symmetric key, store the encrypted data file at a location in the computer file system, and store in the directory an entry containing an encryption of the name and an encryption of the location, wherein the symmetric key is generated in part by dividing a first key into sub-keys each corresponding to a block of the data file, modifying each of the sub-keys based on an identifier of a corresponding block to produce modified sub-keys, and combining the modified sub-keys.

Claim 37 (previously presented): The method of claim 36, wherein the symmetric key encrypts clear data to generate cipher data according to a block cipher.

Claim 38 (original): The method of claim 37, wherein the block cipher comprises a Rijndael algorithm.

Claim 39 (original): The method of claim 37, wherein the block cipher comprises an algorithm selected from the group consisting of DES, triple-DES, Blowfish, and IDEA.

Claim 40 (previously presented): The method of claim 36, wherein executing kernel code comprises:

entering a pass key and a data file name into a first encryption process to produce an encrypted data file name and an encrypted data file name key; and  
processing the clear data file together with the encrypted data file name key to generate an encrypted file contents key and an encrypted file contents.

Claim 41 (previously presented): The method of claim 40, further comprising:

storing the encrypted data file name key and the encrypted file contents key in a first protected area of a computer storage; and  
storing the encrypted data file name and the encrypted file contents in a second protected area of the computer storage.

Claim 42 (previously presented): The method of claim 36, wherein executing kernel code to encrypt the clear data file is performed when data is transferred between a computer memory and a secondary device.

Claim 43 (original): The method of claim 42, wherein the secondary device comprises a backing store.

Claim 44 (original): The method of claim 42, wherein the secondary device comprises a swap device.

Claim 45 (previously presented): The method of claim 42, wherein the secondary device forms part of a network of devices.

Claim 46 (canceled).

Claim 47 (previously presented): The method of claim 45, wherein the network comprises the Internet.

Claim 48 (previously presented): A computer system comprising:

- a. a processor;
- b. a physical memory containing an encrypted data file and a directory, wherein the directory comprises a record having a first element corresponding to an encrypted name of the data file and a second element corresponding to an encrypted location of the data file in the memory;
- c. a secondary device coupled to the physical memory; and
- d. an operating system comprising a kernel, the kernel comprising a virtual node integrated with drivers to directly decrypt the first and second elements to access the encrypted data file from memory when transferring the data file from the memory to the secondary device and to directly re-encrypt the first and second elements when transferring the data file from the secondary device to the memory, wherein the drivers decrypt and re-encrypt the first and second elements using one or more keys generated from identifiers of one or more of the data file, a root directory containing the data file, and a file system containing the root directory.

Claim 49 (previously presented): The computer system of claim 48, wherein the kernel is to encrypt and decrypt data using a symmetric key encryption algorithm.

Claim 50 (original): The computer system of claim 49, wherein the symmetric key encryption algorithm is based on a block cipher.

Claim 51 (previously presented): The computer system of claim 50, wherein the symmetric key encryption algorithm comprises Rijndael algorithm.

Claim 52 (original): The computer system of claim 51, wherein the kernel comprises a UNIX operating system.

Claims 53-58 (canceled)

Claim 59 (previously presented): The computer system of claim 1, wherein the kernel is also to encrypt or decrypt a data file in the directory with a corresponding one of multiple file encryption keys and to encrypt or decrypt the directory with a directory encryption key.

Claim 60 (previously presented): The computer system of claim 59, wherein the multiple file encryption keys are different from each other.

Claim 61 (previously presented): The computer system of claim 1, wherein the encrypted directory comprises encrypted directory information including file names and locations of data blocks.

Claim 62 (previously presented): The computer system of claim 1, wherein the encrypted directory comprises encrypted directory information including data file names and corresponding i-node entries.

Claim 63 (previously presented): The computer system of claim 26, wherein the operating system kernel is also to locate a target directory by comparing an encrypted name of the target directory with encrypted names of candidate directories on the computer system.

Claim 64 (previously presented): The computer system of claim 26, wherein the directory information comprises data file names and locations of data blocks.

Claim 65 (previously presented): The computer system of claim 26, wherein the directory information comprises data file names and corresponding i-node entries.

Claim 66 (previously presented): The method of claim 36, wherein the directory comprises encrypted directory information including data file names and locations of data blocks.

Claim 67 (previously presented): The method of claim 36, wherein the directory comprises encrypted directory information including data file names and corresponding i-node entries.

Claim 68 (previously presented): The computer system of claim 48, wherein the directory comprises data file names and locations of data blocks.



Claim 69 (previously presented): The computer system of claim 48, wherein the directory comprises data file names and corresponding i-node entries.

Claim 70 (previously presented): A computer system containing an operating system, the computer system comprising:

- a kernel comprising a virtual node integrated with drivers to encrypt and decrypt data transferred between a memory and a secondary device, wherein the kernel comprises an encryption engine to encrypt clear data to generate cipher data, the encryption engine also to decrypt the cipher data to generate the clear data;
- a memory coupled to the encryption engine to store the cipher data, wherein the memory comprises a first logical protected memory to store encrypted file data and a second logical protected memory to store encrypted key data;
- an encryption key management system to control access to the encrypted file data and the encrypted key data, wherein the encryption key management system comprises a key engine to receive a pass key and the file name to generate an encrypted file name key, use the encrypted file name key and file contents to generate an encrypted file contents key, and encrypt the file contents with the encrypted file contents key to generate encrypted file contents.

Claim 71 (previously presented): A method of encrypting data, the method comprising:

- receiving clear data; and
- executing kernel code in an operating system, wherein the kernel code comprises a virtual node integrated with drivers to use a symmetric key to encrypt the clear data to generate cipher data and to use the symmetric key to decrypt the cipher data to generate the clear data, and further wherein executing the kernel code comprises entering a pass key and a file name into a first encryption process to produce an encrypted file name and an encrypted file name key and processing the clear data together with the encrypted file name key to generate an encrypted file contents key and encrypted file contents.

Claim 72 (previously presented): The computer system of claim 1, further comprising a plurality of different encryption keys to decrypt corresponding blocks of the data file.

Claim 73 (previously presented): A computer system comprising a memory portion containing an encrypted data file and an operating system comprising a kernel, wherein the kernel comprises a virtual node to decrypt an encrypted directory entry to determine a location of the encrypted data file and to decrypt the encrypted data file to access data contained therein, the virtual node to decrypt the data file using a first key generated from an identifier of the operating system, an identifier of a file system containing the data file, an identifier of a root directory containing the encrypted data file, an identifier of the data file, and a second key.

Claim 74 (previously presented): A computer system comprising:  
a memory portion containing an encrypted data file, a first logical protected memory to store encrypted data files and a second logical protected memory to store encrypted key data, and an operating system comprising a kernel, wherein the kernel comprises a virtual node (a) to directly decrypt an encrypted directory entry to determine a location of the encrypted data file and (b) to directly decrypt the encrypted data file to access data file contents contained therein; and  
an encryption key management system to control access to the encrypted data files and the encrypted key data, wherein the encryption key management system comprises a key engine, the key engine to receive a pass key and a data file name to generate an encrypted data file name key, the key engine also to use the encrypted data file name key and the data file contents to generate an encrypted data file contents key, the key engine also to encrypt the data file contents with the encrypted data file contents key to generate encrypted data file contents and to encrypt the data file name with the encrypted data file name key to generate an encrypted data file name.

Claim 75 (previously presented): A method of storing an encrypted data file in a computer file system having a directory, the method comprising:  
receiving a clear data file having a name; and  
executing kernel code in an operating system, the kernel code comprising a virtual node comprising drivers to directly encrypt the clear data file to generate an encrypted data file using a symmetric key, store the encrypted data file at a location in the computer file system, and store in the directory an entry containing an encryption of the name and an encryption of the location, wherein executing kernel code comprises:

entering a pass key and a data file name into a first encryption process to produce an encrypted data file name and an encrypted data file name key; and processing the clear data file together with the encrypted data file name key to generate an encrypted file contents key and an encrypted file contents.